Scope Rules and Binding in Comal80

Written for the Comal80 Development Group
by Arne Christensen, Metanic ApS. March 1983

The following points show Metanic's position on scope rules and binding in Comal80:

1. Static binding is used everywhere. It must be clear from the program text what each name stands for. This is reflected by several of the rules below.
2. All variables introduced in a Closed procedure or function are local to the procedure or function and cannot be referenced from within any other procedure or function from the main program. This also applies to the parameters of the Closed procedure/function since these are instances of local variables.
3. From within a Closed procedure/function the following variables can be referenced: The local variables of the procedure/function and the variables explicitly imported from the main program by means of the IMPORT statement. No other variables can be referenced from within the Closed procedure/function.
4. Parameters to an Open procedure/function are local to the procedure/function and cannot be referenced from within any other procedure/function (not even from within another Open procedure/function) or from the main program. An Open procedure/function has no local variables apart from the parameters.
5. From within an Open procedure/function the following variables can be referenced: The parameters of the procedure/function and the variables of the main program. No other variables can be referenced from within the Open procedure/function. If any variables are DIMensioned within the Open procedure, they are in all respects treated as if they were DIMensioned within the main program. Also, variables that are used within the Open procedure/function but not in the main program, are still considered belonging to the main program.
6. The IMPORT statement can be used in Closed procedures/functions only. It cannot be used in an Open procedure/function or in the main program.
7. System variables and functions can be referenced everywhere.
8. Procedures and functions cannot be nested, whether they are Open or Closed.
9. Procedures and functions can call each other and themselves recursively, whether they are Open or Closed.
10. Procedures can be called throughout the program. Procedure names cannot be IMPORTed and there is no need to do so.
11. Functions can be called throughout the program except at points where a variable with the same name hides the function. This is only possible if the function has parameters. The rule is: If a DIMensioned variable with the same name as the function can be referenced at a certain point then the function cannot be called at this point. Any attempt to call the function at this point will be interpreted as a reference to the variable with the parameters interpreted as indices.

    A call to a parameterless function is programmed by writing the name of the function followed by an empty paranthesis, e.g. FUNCTION(). Thus there is no possibility of confusion even if a variable with the same name can be referenced at the point of the call.

    The rules above are introduced in order to allow for arbitrary naming of local variables in library procedures which will always be Closed procedures.

Function names cannot be IMPORTed and there is no need to do so.

12. Labels defined in the main program can only be referenced in the main program. Labels defined in a procedure or function, whether it be Open or Closed, can only be referenced in the procedure or function where it is defined. Labels cannot be IMPORTed.